CLAIMS

1. A method of transferring data from an I/O node to a host across a channel-based switching fabric interconnect, the method comprising:

storing a value in a register in the I/O node which is indicative of a number of send credits available to the I/O node;

determining from the value of the register if there is a sufficient number of send credits available to the I/O node for the data to be transferred;

promptly transferring the data from the I/O node to the host over the channel-based switching fabric interconnect if a sufficient number of send credits is available to the I/O node; and

otherwise, if a sufficient number of send credits is not available to the I/O node, waiting for the host to update the value stored in the register before transferring data.

2. The method of claim 1, wherein the data is transferred by a send message sent from the I/O node to the host over the channel-based switching fabric when sufficient send credits are available for the data.

3. The method of claim 1, wherein each of the send credits available to the I/O node represent one or more receive buffers which are available at the host for receiving and storing a data packet.

4. The method of claim 2, wherein said step of transferring data by sending a send message comprises:

preparing a descriptor describing the send operation to be performed; and

posting the descriptor to one of a plurality of work queues in the I/O node.

5. The method of claim 4, wherein the I/O node further comprises a plurality of send buffers storing the data to be transferred, the step of transferring data by sending a send message comprises:

preparing a descriptor describing the send operation to be performed;

posting the send descriptor to one of the work queues in the I/O node;

processing the posted send descriptor by transferring the data from one of the send buffers to the channel-based switching fabric interconnect.

6. The method of claim 4, wherein the host updates the value stored in the register by performing an RDMA write operation to the register.

7. The method of claim 5, further comprising:

fragmenting the data to be transferred into two or more data packets;

performing the following steps until all data packets have been sent:

a) determining if a sufficient number of send credits is available at the I/O node;

b) sending a data packet from the I/O node over the channel-based switching fabric if a sufficient number of send credits is available, and adjusting the number of send credits based on the sending of the data packet; and

c) otherwise, if a sufficient number of send credits is not available at the I/O node, waiting for the host to update the value stored in the register before sending a data packet.

8. An I/O node configured to communicate with a host across a channel-based switching fabric interconnect, the I/O node comprising:

a channel adapter connecting the I/O node to the channel-based switching fabric; and

a virtual interface, including:

a plurality of send and receive buffers;

a transport service layer, the transport service layer transferring data between the I/O node and the host;

an interface user agent coupled to the transport service provider;

a kernel agent;

a plurality of work queues; and

a network interface controller coupled to the kernel agent, the work queues and the channel adapter;

said virtual interface to issue one or more control commands to the kernel agent to establish a connection between the I/O node and the host across the channel-based switching fabric and to post data transfer requests to the work queues in response to commands from the transport service layer; and

the network interface controller to process the data transfer requests by transferring data between the send and receive buffers and the channel adapter.

9. The I/O node of claim 8 wherein the virtual interface is in accordance with at least a portion of the Virtual Interface (VI) Architecture.

10. The I/O node of claim 9, wherein the kernel agent comprises a Virtual Interface (VI) kernel agent, and the network interface controller comprises a Virtual Interface (VI) network interface controller.

11. An I/O node configured to communicate with a host across a channel-based switching fabric, said I/O node comprising:

a memory including send and receive application buffers;

a transport service layer providing for data transfer across the channel-based switching fabric;

a network interface controller coupled to the network;

a plurality of work queues coupled to the network interface controller for posting data transfer requests thereto;

a user agent coupled to the send and receive buffers and the network interface controller, the user agent posting data transfer requests to the work queues, the network interface controller processing the posted data transfer requests by transferring data between the send and receive buffers and the channel-based switching fabric.

12. An I/O node configured to communicate with a host computer over a channel-based switching fabric interconnect, the I/O node comprising:

a processor;

a register storing send credits;

one or more work queues for posting data transfer requests;

one or more registered send buffers;

one or more registered receive buffers;

a network interface controller coupled to the processor, the work queues, the buffers and the channel-based switching fabric, the network interface controller processing the posted data transfer requests by transferring data between the registered buffers and the channel-based switching fabric interconnect; and

the processor being programmed to control the transfer of data through the network interface controller according to a credit-based flow control scheme depending on the send credits stored in said register.

13. The I/O node of claim 12, wherein said processor is programmed to perform the following:

determine if a sufficient number of send credits is available;

send a send message containing a data packet from the I/O node to the host over the channel-based switching fabric interconnect if a sufficient number of send credits are available; and

otherwise, if a sufficient number of send credits is not available, waiting for the host to update the value stored in the register before transferring the data packet.

14. The I/O node of claim 13, wherein the I/O node places a count value of the number of data transfers in the send message.

15. The I/O node of claim 13, wherein the host updates the value stored in the register by performing an RDMA write operation to the register of the I/O node.

16. A host computer comprising:

a network interface controller adapter connecting the host to a host channel adapter on a channel-based switching fabric interconnect;

a host processor;

a memory having registered send and receive buffers; and

a device driver coupled to the host processor and the memory, and having one or more work queues for posting data transfer requests and a transport service layer providing an end-to-end credit-based flow control across the channel-based switching fabric interconnect according to the status of said registered receive buffers.

17. The host computer recited in claim 16, wherein the device driver comprises a virtual interface, including:

a transport service layer, the transport service layer transferring data between the I/O node and the host;

an interface user agent coupled to the transport service provider; and

a kernel agent coupled to the the kernel agent and the work queues,

said virtual interface issuing one or more control commands to the kernel agent to establish a connection between the I/O node and the host across the channel-based switching fabric and posting data transfer requests to the work queues in response to commands from the transport service layer.

18. The host computer of claim 17, wherein the virtual interface is in accordance with at least a portion of the Virtual Interface (VI) Architecture.

19. The host computer of claim 18, wherein the kernel agent comprises a Virtual Interface (VI) kernel agent, and the network interface controller comprises a Virtual Interface (VI) network interface controller.

20. The host computer of claim 16, wherein the device driver allocates receive buffers in the memory and performs an RDMA write operation to the I/O node to update a register storing the send credits of said I/O node.